UE23CS251B: Microprocessor and Computer Architecture (4-0-2-5-5)

This course will give you an in-depth understanding of the inner-workings of modern digital computer systems and trade-offs present at the hardware-software interface. The course focuses on key topics in microprocessor such as the system architecture, low level programming aspects and interface with other key components. Also, the course will help in understanding the core computer architecture concepts such as multilevel in memory hierarchies, pipelining, and super scalar techniques. A desirable knowledge of Digital Design and Computer organization is required.

Course Objectives:
- Introduce concepts of basic processor architecture and its design.
- Understanding the concept of concepts of pipeline architecture and hazards.
- Study of memory hierarchy, cache memory and its optimizations.
- Introduce advanced concepts in processor architecture like multi-core/ many core processor architectures.

Course Outcomes:
At the end of the course, the student will be able to:
- Demonstrate ability to understand the design of different instruction sets like RISC/ CISC and their addressing modes.
- Demonstrate the ability to understand the design of a pipelined processor and its challenges.
- Demonstrate the use of tools to analyse the performance of programs on different architectures. Design alternative memory hierarchy layouts and optimizations.
- Demonstrate and appreciate modern trends in architecture such as multicore architectures.

Desirable Knowledge: UE23CS251A- Digital Design and Computer Organization.

Course Content:
Unit 1 : Architecture
Introduction, ISA Classification - RISC and CISC, Memory Addressing, Operands - Types and Size, Instruction Set - Operations, Control Flow, Instruction Encoding, Case Study - ARM/ MIPS/ x86 Processor.

14 Hours

Unit 2 : Pipelining
3- Stage Pipelining, 5 - Stage Pipelining, Pipeline Datapath and Control, Data Hazards – Forwarding vs. Stalling, Control Hazards, Branch Prediction Mechanisms and Exceptions, Performance Metrics.

14 Hours

Unit 3 : Basics of Cache and Cache Optimization
Basics of Caches - Fully Associative, Direct Mapped and Set Associativity, Cache Performance, Basic Cache Optimization- Reduce in Miss Rate. Basic Cache Optimization- Reduce Miss Penalty, Reduce Hit Time.

14 Hours

Unit 4 : Advances in Architecture
Introduction to Parallel Computing, PC – Applications, Memory architecture, Flynn’s taxonomy, parallel programming models, Shared memory programming OpenMP-Introduction, loop-level parallelism, CUDA C Program structure-vector kernel addition, device global memory and Data transfer, Hardware Multi threading, Parallel examples: matrix multiplication, PC-Design Issues, Amdahl’s Law, Gustafson Law, Multi-Core Architecture, Introduction to GPU computing

14 Hours

Laboratory :
Course Content:
1. Introduction to ARM Simulator and sample programs
2. Implementation of Data Processing Instructions, Programs on usage of addressing modes
3. Programs on functions and software interrupts, Matrix Operations – Addition and Multiplication
4. Introduction to PARACACHE simulator – Direct Mapping, Associative Mapping
5. Introduction to Plugins interface.
6. Project work

Tools/ Languages: ARM Simulator, Arduino Microcontroller kit, ParaCache simulator.

Text Book(s):

1: "Computer Organization and Design", Patterson, Hennessey, 5th Edition, Morgan Kaufmann, 2014.
2. "Computer Organization and Design – ARM Edition", Patterson, Hennessey, 4th Edition, Morgan Kaufmann, 2010.
3. "ARM System-on-Chip Architecture", Steve Furber, 2nd Edition, Pearson India, 2015.
4. "Programming Massively Parallel Processors A Hands-on Approach", Third Edition
David B. Kirk Wen-mei W. Hwu, Morgan Kaufmann
5. "Multicore and GPU Programming An Integrated Approach", Gerassimos Barlas, second edition, Morgan Kaufmann

Reference Book(s):

1: "Computer Architecture: A Quantitative Approach", Hennessey, Patterson, 5th Edition, Morgan Kaufmann, 2011.
2: "The Definitive Guide to the ARM Cortex-M0 and Cortex MO+ processors", Joseph Yiu, 2nd Edition, Newnes, 2015

UE23CS252B: Computer Networks (4-0-2-5-5)

This is a foundation course on Computer Networking which focuses on building blocks of the Internet. We trace the journey of messages sent over the Internet from an application residing on one host machine (source) to another (Destination) using a top-down layered approach. The course contents are organized based on TCP/IP Protocol stack.

Course Objectives:

- To present a broad overview of computer networking, the Internet and network layered architecture.
- To study the conceptual and implementation aspects of network applications, protocols and socket programming.
- To provide an insight of the Internet's connection–oriented and connectionless end–to–end transport service protocols and TCP's approach to congestion control, to learn exactly how the network layer can provide its host–to–host communication service.
- To study the IPv4 and IPv6 protocols, to explore several important link–layer concepts and technologies, LAN and Wireless LANs.

Course Outcomes:

At the end of this course, the student will be able to:
- Demonstrate in a concise way how the Internet is constructed and functions with respect to TCP/IP or OSI reference models.
- Demonstrate application layer protocols like DNS, HTTP, HTTPS and be able to develop simple client–server applications using socket programming and understand the concept of unreliable data transfer protocols and how UDP implement these concepts.
- Understand the concept of reliable data transfer protocols and how TCP implement these concepts. Implement logical addressing schemes and configure devices using NAT.
- Demonstrate the ability to configure the routers and services such as DHCP, ICMP. Construct and troubleshoot a wired or wireless LAN, and be able to understand wider networking issues.

Course Content:

Unit 1- Computer Networks and the Internet, Application Layer
Introduction to Computer Networks, Internet: A Nuts–and–Bolts Description, A Services Description, Protocol, The Network Edge: Access Networks, Physical Media, Introduction to physical layer devices, The Network Core, Packet Switching, Circuit Switching, A Network of Networks, Delay, Loss, and Throughput in Packet–Switched Networks, Overview of Delay in Packet Switched Networks – Queuing Delay and Packet Loss, End–to–End Delay, Throughput in Computer Networks, The OSI Model and the TCP/IP Protocol Suite, Protocol Layers, The OSI Model, TCP/IP Protocol Suite. Network Application Principles: Network Application Architectures, Processes Communication, Transport Services available to Applications, Transport Services provided by Internet.

14 Hours

Unit 2 - Application Layer, Transport Layer – UDP
The Web, HTTP and HTTPS, Non persistent and persistent connection, HTTP Message Format, User Server Interaction: Cookies, Web Caching. DNS, The Internet's Directory Service: Services provided by DNS; How DNS works: DNS Records and messages; Peer to peer Applications; Socket Programming with TCP and UDP; Other Application Layer Protocols: FTP, SMTP, SNMP, Telnet, SSH.
Introduction to Transport Layer Services: Relationship Between Transport and Network Layer, Overview of the Transport layer in the Internet, Multiplexing and Demultiplexing; Connectionless Transport UDP: UDP Segment Structure, UDP Checksum.

14 Hours

Unit 3- Transport Layer – TCP, Network Layer and Internet Protocol
Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocol, Go–Back–N Protocol, Selective–Repeat; Connection Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Flow Control, TCP Connection Management, TCP Congestion Control. Numerical on TCP congestion control mechanisms–TCP Tahoe, Reno.
Overview of Network Layer: Forwarding and routing, what's Inside a Router? The Internet Protocol (IP) IPV4: Datagram Format, Fragmentation, Addressing, NAT.

14 Hours

Unit 4 - Network Layer and Internet Protocol, Link Layer and LAN

Introduction to Network layer Protocols: DHCP, ICMP; IPv6 Protocol: Packet Format, Transition from IPv4 to IPv6; Introduction to Routing Algorithms: Link State: Dijkstra's algorithm and Distance Vector: Bellman–Ford Algorithm.
Link layer – Error–Detection and Correction techniques, Parity checks, Internet Checksum, Cyclic Redundancy Check, and Multiple Access Protocols: CSMA/CD, CSMA/CA; Switched LAN: Link layer addressing and ARP, Ethernet: Link–layer switches. Retrospective: A Day in the Life of a Web Page Request. Physical Layer – Purpose, Signals to Packets, Transmission media. Wireless LANs: IEEE 802.11 LAN architecture, 802.11 MAC Protocol, IEEE 802.11 Frame.

14 Hours

Tools: Wireshark, Python.

Text Book(s):
1: "Computer Networking: A Top – Down Approach", James F. Kurose, Keith W. Ross, 7th Edition, Pearson Publication, 2017.

Reference Book(s):
1: "TCP IP Protocol Suite", Behrouz Forouzan, 4th Edition, McGraw–Hill, 2010.

Lab/ Hands-on: **14 Hours**

1. Program on ping, tcpdump and wireshark.
2. Program on Exploring HTTP with wireshark, Web Server setup, FTP/ SMTP and SNMP Clients, Telnet, SSH and DNS
3. Program on Wireshark based TCP congestion window plotting, UDP traffic analysis.
4. Program on Cisco Packet Tracer based Router experiments; IPv4 Fragmentation based Wireshark experiments, Inspection of DHCP, ICMP.
5. Program on Wireshark based Link Layer protocol inspection.

UE23CS241B - Design and Analysis of Algorithms (4-0-0-4-4)

Algorithms play a key role in science and practice of computing. Learning algorithm design technique is a valuable endeavour from practical standpoint and algorithm design techniques have considerable utility as general problem solving strategies, applicable to problems beyond computing. This course includes classic problems of computer science, application of design techniques and analysis in terms of time and space.

Course Objectives:

- Learn various algorithm design techniques and apply appropriate algorithmic design techniques for specific problems.
- Learn to design and analyze algorithms with an emphasis on resource utilization in terms of time and space
- Learn to trade space for time in algorithmic design using input enhancement and per-structuring.
- Learn the limitations of algorithmic power and techniques to handle these limitations

Course Outcomes:

At the end of the course, the student will be able to:

- Identify the design technique used in an algorithm
- Design and implement efficient algorithms for practical and unseen problems and analyze these algorithms using quantitative evaluation.
- Analyse time efficiency over trading space
- Understand the limits of algorithms and the ways to cope with the limitations.

Course Content:

Unit 1: Introduction and Brute Force

Algorithms, Fundamentals of Algorithmic Problem Solving, Important Problem Types. Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non Recursive and Recursive Algorithms. Brute Force: Selection Sort, Bubble Sort, Sequential Search, Brute Force String Matching, Exhaustive Search.

14 Hours

Unit 2: Decrease – and – Conquer & Divide-and-Conquer

Decrease-and-Conquer: Decrease by constant number algorithms - Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms – Fake Coin Problem, Russian Peasant Multiplication, Josephus problem, Decrease-by-Variable-Size Algorithms – Computing a median and the selection problem. Divide-and-Conquer: Master Theorem, Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Complexity analysis for finding the height of BST, Multiplication of Large Integers, Strassen's Matrix Multiplication.

14 Hours

Unit 3: Transform-and-Conquer Space and Time Tradeoffs & Greedy Technique

Transform and Conquer**:** Pre-sorting, Heap Sort, Red-Black Tree Construction and Time complexity Analysis for insert and search operation, 2-3 Trees and B Tree: insertion, deletion, searching, and time complexity analysis. Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching - Horspool's and Boyer-Moore Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm and union-find algorithm, Dijkstra's Algorithm, Huffman Trees

14 Hours

Unit 4: Limitations, Coping with the Limitations of Algorithm Power & Dynamic Programming,

Dynamic Programming: Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. Limitations of Algorithm Power**:** Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete, NP-Hard Problems. Coping with the Limitations of Algorithm Power: Backtracking, Branch-and-Bound.

14 Hours

Tools / Languages: C Programming Language, GCC Compiler.

Text Book(s):

1: "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi (Indian Version), 3rd Edition, 2012.

Reference Book(s):

1: “Introduction to Algorithms”, Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, Prentice-Hall India, 3rd Edition, 2009.
2: “Fundamentals of Computer Algorithms”, Horowitz, Sahni, Rajasekaran, Universities Press, 2nd Edition, 2007.
3: “Algorithm Design”, Jon Kleinberg, Eva Tardos, Pearson Education, 1st Edition, 2006.

UE23CS242B: Operating Systems (4-0-0-4-4)

This course focuses on fundamental operating systems concepts including various algorithms and trade-offs for efficient management of resources such as CPU, Memory, Storage and I/O. This course requires the student to have a desirable knowledge of Data Structures and its Applications.

Course Objectives:

- Learn the fundamental Operating System concepts and various algorithms for process management.
- Demonstrate various algorithms and associated trade-offs for efficient resource management such as inter process communication, threads, process synchronization, deadlocks.
- Provide an understanding and apply various memory management techniques.
- Understand file system and secondary storage structures.

Course Outcomes:

At the end of the course, the student will be able to:

- Understand the design of various algorithms for scheduling and their relative performance.
- Understand and apply Inter Process Communication, threads, process synchronization and deadlocks
- Apply various memory management techniques.
- Understand file system, its implementation and the various secondary storage structures.

Desirable Knowledge: UE23CS252A - Data Structures and its Applications.

Course Contents:

Unit 1 : Introduction and Process Management

What Operating Systems Do, Operating-System Structure & Operations, Kernel Data Structures, Operating-System Services, Operating System Design and Implementation.

Shell programming: Overview of bash shell programming – variables, control flow

Processes: process concept, Process Scheduling, Operations on Processes, System calls for process management-fork (), vfork (), wait () and exec ().

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms. Case Study: Linux Scheduling Policies.

Shell programming – cron.

14 Hours

Unit 2 : IPC, Threads and Concurrency

IPC: Introduction, Shared Memory systems, Message Passing, Communication in Client–Server Systems- Pipes, ordinary pipes and named pipes, system calls for shared memory, pipes and FIFO's.

Threads: Overview, Multicore Programming, Multithreading Models, Thread Libraries, Thread Scheduling.

Process Synchronization: Background, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization- The Bounded-Buffer Problem, The Readers–Writers Problem, The Dining-Philosophers Problem, Synchronization Examples- Synchronization in Linux. System calls for threads creation and synchronization-POSIX Threads.

Deadlocks: System Model, Deadlock Characterization, Deadlock avoidance, Banker's Algorithm, Deadlock Detection, Deadlock Recovery.

14 Hours

Unit 3: Memory Management

Main Memory: Background- Basic Hardware, Address Binding, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking and Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement Algorithms-FIFO, LRU, Optimal, Allocation of Frames, Thrashing.

14 Hours

Unit 4 : File and Storage Management

File-System Interface: File Concept, system calls for file operations-open(), read(),write(), lseek(), close() and system call to retrieve file attributes and file types-stat(), lstat(), Access Methods, Directory and Disk Structure, system calls for reading directories, system calls to create hard links (link()) and symbolic links-symlink().

File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, File Sharing, Protection.

Storage management: Overview of Mass-Storage Structure, Disk Scheduling, Swap-Space Management, RAID Structure.

System Protection: Goals, Principles and Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control; **System Security**
Shell programming - awk, sed

14 Hours

Tools/Languages/OS : C, Linux/Unix OS for system call implementation.

Text Book(s):
1: “Operating System Concepts”, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 9th Edition, John Wiley & Sons, India Edition ,2016.
2: “Advanced Programming in the Unix Environment”, Richard Stevens and Stephen A Rago, Pearson, 3rd edition, 2017.
3: “Your UNIX/Linux: The Ultimate Guide”, Sumitabha Das, 3rd Edition, McGraw-Hill, 2013.

Reference Book(s):
1: “Operating Systems, Internals and Design Principles”, William Stallings, 9th Edition, Pearson, 2018.
2: “Modern Operating Systems”, Andrew S Tanenbaum, 3rd edition, Pearson, 2007.
3: “Learning the bash shell”, Cameron Newham, 3rd edition, O’Reilly, 2005

UE23CS243B - Linear Algebra and its Applications (4-0-0-4-4)

This is a basic subject on matrix theory and linear algebra. Emphasis is given to topics that will be useful in computer science discipline, including systems of equations, vector spaces, eigenvalues, similarity, and positive definite matrices. The course provides hands-on experience in basic programming concepts using PYTHON/R for solving problems relevant to these areas.

Course Objectives:

- The first goal of the course is to teach students how to use linear algebra as a powerful tool for computation.
- The second goal is to show how these computations can be conceptualized in a geometric framework.
- The third goal is to give a gentle introduction to the theory of abstract vector spaces.
- To visualize solution to linear system of equations with different approaches using Python.

Course Outcomes:

After completing this course, students will be able to:

- Solve systems of Linear Equations using Matrix Transformations, Interpret the nature of Solutions, Visualize Consistency of Linear system of Equations and also compute inverse of a Matrix.
- Demonstrate the ability to work within Vector Spaces, distil Vector Space properties and understand the concepts of the four fundamental Subspaces, Linear Span, Linear Independence, Dimension and Basis
- Learn the concepts of Orthogonal Vectors and Orthogonal Subspaces and apply the Gram-Schmidt process to find an Orthonormal Basis in a Subspace, Eigenvalues, Eigenvectors and Diagonalization of a Matrix.
- Apply the concept of Positive Definite Matrices, Singular Value Decomposition into Application problems.

Course Content:

Unit 1: Matrices, Gaussian Elimination and Vector Spaces

Introduction, The Geometry of Linear Equations, Gaussian Elimination, Singular Cases, Elimination Matrices, Triangular Factors -LU decomposition &**Cholesky's method**and Row Exchanges, Inverses and Transposes, Inverse by Gauss -Jordan method, Vector Spaces and Subspaces (definitions only)

Application:

1: Basic operations with matrices in Matlab
2: Matrix operations and image manipulation
3: Matrix multiplication, inversion, and photo filters
4: Solving linear systems

Self-Learning Component: Algebra of Matrices.

14 Hours

Unit 2: Four Fundamental Subspaces & Linear Transformations

Linear Independence, Basis and Dimensions, Row reduced Echelon form, The Four Fundamental Subspaces, Rank-Nullity theorem. Linear Transformations, Algebra of Linear transformations,

Application:

1: Systems of linear equations and college football team ranking (with an example of the Big 12)
2: Convolution, inner product, and image processing revisited
3: Norms, angles, and your movie choices
4: Interpolation, extrapolation, and climate change

Self-Learning Component: Examples of Vector Spaces and Subspaces.

14 Hours

Unit 3: Orthogonalization, Eigenvalues and Eigenvectors

Orthogonal Vectors and Subspaces, Orthogonal Bases, Cosines and Projections onto Lines, Projections and Least Squares. Orthogonalization, The Gram-Schmidt Orthogonalization process, Introduction to Eigenvalues and Eigenvectors, Properties of Eigenvalues and Eigenvectors, Cayley-Hamilton theorem (statement only), Diagonalization of a Matrix

Applications:

1. Orthogonal matrices and 3D graphics
2. Discrete dynamical systems, linear transformations of the plane, and the Chaos Game
3. Projections, eigenvectors
4. Matrix eigenvalues and the Google's PageRank algorithm

14 Hours

Unit 4: Singular Value Decomposition

Symmetric Matrices, Quadratic Forms, Definitions of Positive definite, negative definite, positive semi-definite, negative semi-definite, indefinite forms and matrices, Tests for Positive Definiteness, Singular Values and Singular Vectros, Image Processing by Linear Algebra, Principal Component Analysis(PCA by the SVD), Minimizing a Multivariate Functions, Back Propagation and Stochastic Gradient Descent

Applications:

1. Principal Component Analysis, and face recognition algorithms
2. Social networks, clustering, and eigenvalue problems
3. Singular Value Decomposition and image compression

14 Hours

Tools ' Languages : Python IDE

Text Book(s):

1.Linear Algebra and its Applications, Gilbert Strang, Thomson Brooks/ Cole, 5th Edition, Second Indian Reprint 2007.

2. Application : Applied projects for an introductory linear algebra class by Anna Zemlyanova

Reference Book(s):

1. Linear Algebra and its Applications, David. C lay, Publication by Pearson Education, 5th Edition, 2015
2. Linear Algebra, Schaum's outlines, Seymour Lipschutz and Marc Lipson, Tata McGraw-Hill publications, 4thEdition, 2009.
3. Higher Engineering Mathematics, B S Grewal, Khanna Publishers, 44th Edition, 2020,.
4. Practical Linear Algebra, Gerald Farin and Dianne Hansford, CRC Press, Taylor & Francis Group, 3rd Edition, 2013.